Bioimage informatics

CellProfiler Analyst 3.0: accessible data exploration and machine learning for image analysis

David R. Stirling, Anne E. Carpenter and Beth A. Cimini

Imaging Platform, Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA

*To whom correspondence should be addressed.

Abstract

Summary: Image-based experiments can yield many thousands of individual measurements describing each object of interest, such as cells in microscopy screens. CellProfiler Analyst is a free, open-source software package designed for the exploration of quantitative image-derived data and the training of machine learning classifiers with an intuitive user interface. We have now released CellProfiler Analyst 3.0, which in addition to enhanced performance adds support for neural network classifiers, identifying rare object subsets, and direct transfer of objects of interest from visualization tools into the Classifier tool for use as training data. This release also increases interoperability with the recently released CellProfiler 4, making it easier for users to detect and measure particular classes of objects in their analyses.

Availability: CellProfiler Analyst binaries for Windows and MacOS are freely available for download at https://cellprofileranalyzer.org/. Source code is implemented in Python 3 and is available at https://github.com/CellProfiler/CellProfiler-Analyst/. A sample dataset is available at https://cellprofileranalyzer.org/examples, based on images freely available from the Broad Bioimage Benchmark Collection.

Contact: bcimini@broadinstitute.org

1 Background

With the increasing adoption of high-throughput microscopy, scientists have been able to generate large datasets containing thousands of individual images. This necessitates automated computational analysis to efficiently derive biological insights from the raw data. Free software packages such as ImageJ (Schneider et al., 2012) and CellProfiler (McQuin et al., 2018) allow users to extract hundreds or thousands of numerical measurements from their image data, but it is ultimately on the user to determine which of these features are relevant to the biological problem being investigated. Spreadsheet programs are familiar but lack features and integration with source images that biologists often need. CellProfiler Analyst is a data exploration package for helping users to explore and extract information from large datasets, including those produced by CellProfiler pipelines (Jones et al., 2008). The software includes several tools for users to visualize and filter their datasets, alongside tools for training machine learning classifier models within a convenient graphical user interface that is geared toward working with image data (Dao et al., 2016). While other tools such as Ilastik (Berg et al., 2019) and Advanced Cell Classifier (Piccinini et al., 2017) can provide a GUI for training classifiers with image-based data, these do not include data exploration and visualization tools like those in CellProfiler Analyst. These tools provide an intuitive interface for scientists to explore their data in forms such as histograms and scatter plots, though without the advanced statistical tools or extreme customizability of a pure programming language. Herein we present CellProfiler Analyst 3.0, which includes major performance improvements and new features that improve the utility of the software.

2 General changes

We ported CellProfiler Analyst to the Python 3 programming language to ensure compatibility with future operating systems after the official Python 2 end-of-life in 2020. We also revised the program’s build to package all Java dependencies within the main installer, which dramatically simplifies the installation process. In addition, we added a faster imageio-based image loader to supplement the existing bioformats-based implementation (Silvester et al., 2020). While bioformats provides broader file format compatibility, imageio allows for common image formats to be loaded more efficiently, which greatly improves the time to generate object thumbnails. At present neither of these loaders supports accessing only partial sections of an image, which may be an area for further development to handle large files produced by whole slide imaging.

CellProfiler Analyst 3.0 can export machine learning models that are compatible with CellProfiler 4.2+, allowing the resulting classifiers to be directly embedded into CellProfiler pipelines. This maintains and expands upon the interoperability between the two programs.
objects during database fetching, reducing the loading time for a sample dataset by over 10-fold (Fig. 1B). We also found that removing redundant database calls and unnecessary caching steps during classifier model training reduced processing time by 90% (Fig. 1C).

Database handling improvements were also made in the functions for scoring the datasets after training. These refinements produced a modest improvement in the performance of the ‘score all’ (Fig. 1D) function. We further optimized the ‘score image’ function by revising the workflow for fetching object coordinates, which significantly reduced the time taken to display a scoring preview overlay (Fig. 1E). Usability improvements to the image viewer include a shortcut to resize the image to fit the window and a table display outlining counts of each class found in the image.

5 Future directions

The next major frontier for phenotype classification is to train deep learning models straight from raw pixels rather than requiring a separate feature extraction step (Lucas et al., 2021). Enabling scoring of phenotypes that fall along a continuum rather than into discrete bins would also be useful, as we recently found in red blood cell aging (Doan et al., 2020). A remaining limitation in our software is that only MySQL and SQLite databases can be accessed, though migrating toward using packages such as sqalchemy could expand on compatibility in the future. Ultimately, maintaining CellProfiler Analyst as an up-to-date resource for users to explore high-dimensional, image-based data without needing to code will help the biology community for years to come.

Acknowledgements

The authors would like to thank Pearl Ryder, Erin Weisbart, Jane Hung, David Dao, Egor Zindy and Mario Emmenlauer for contributions to bug fixes and testing of pre-release versions of this software. They also thank all the members of the bioimaging community who have provided feedback and suggestions which have helped to guide this work.

Funding

This work was supported by the National Institutes of Health grants (R35 GM122547 and P41 GM135019 to A.E.C.). This project has been made possible in part by grant number 2020-225720 to B.A.C. from the Chan Zuckerberg Initiative DAF, an advised fund of the Silicon Valley Community Foundation. The funders had no role in study design, data collection and analysis, decision to publish, or manuscript preparation.

Conflict of Interest: none declared.

References